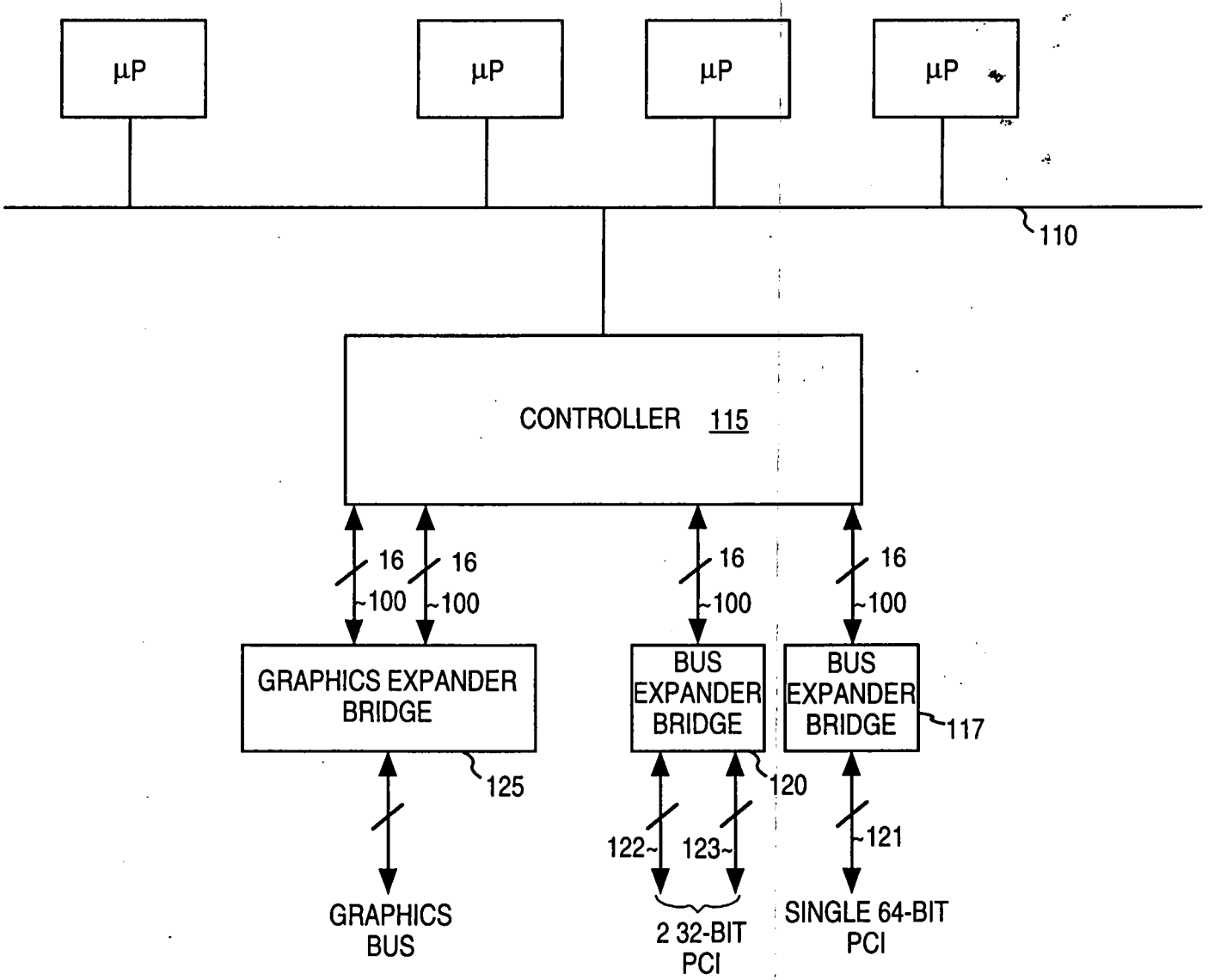

FIG. 1

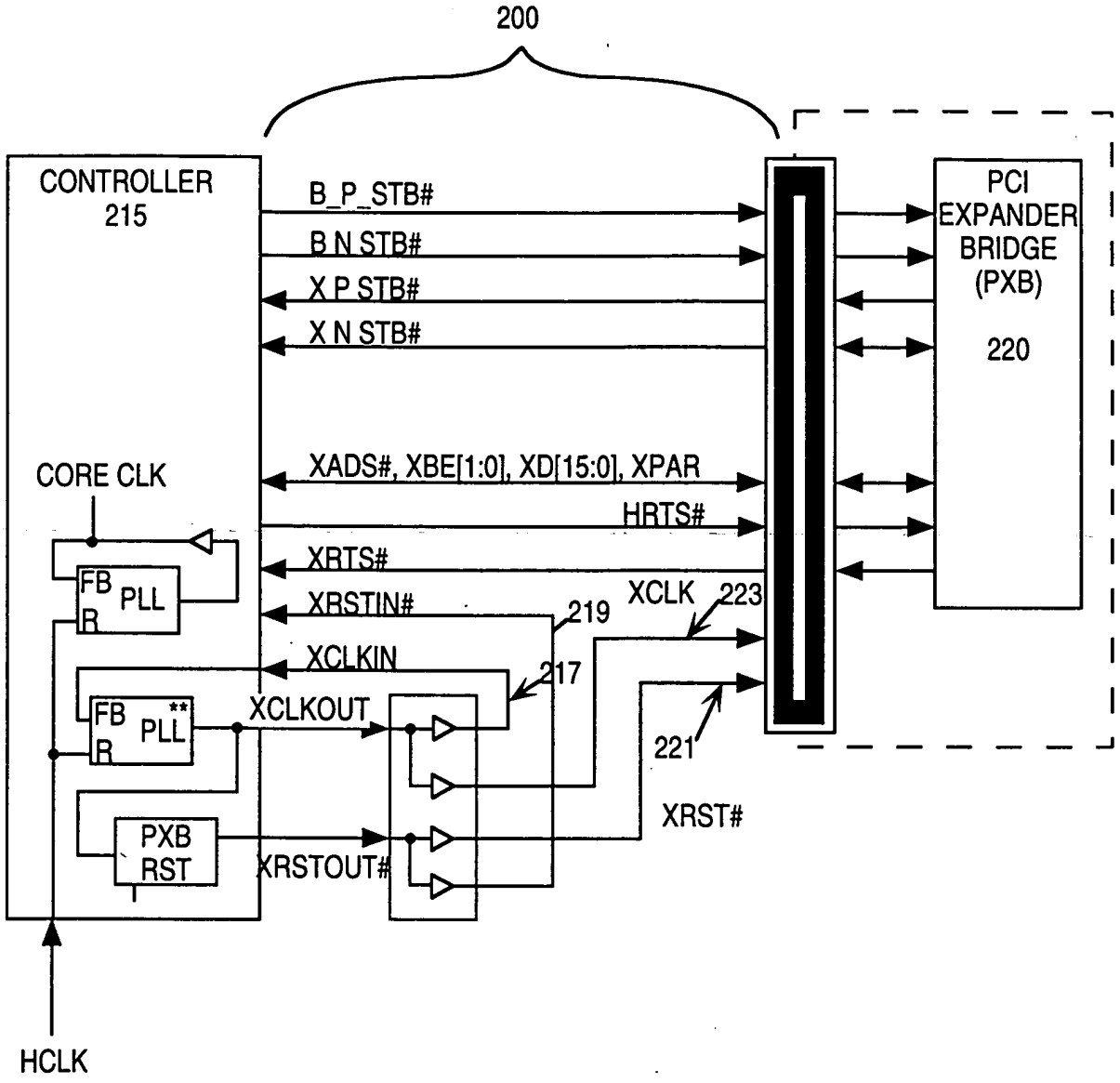
200
CONTROLLER
215
B_P_STB#
B N STB#
X P STB#
X N STB#
PCI
EXPANDER
BRIDGE
(PXB)
220
CORE CLK
XADS#, XBE[1:0], XD[15:0], XPAR
HRTS#
XRTS#
XRSTIN#
XCLKIN
XCLKOUT
XCLK
223
219
217
221
XRST#
XRSTOUT#
FB
R
PLL
**
PXB
RST
HCLK

FIG. 2A

SIGNAL	SRC	DEST.	DEFINITION
XCLKOUT	CONTROLLER	EXT. CLK DRIVER	A COPY OF THE CONTROLLER CORE CLOCK GENERATED BY THE XCLK PLL. AN EXTERNAL LOW SKEW CLOCK DRIVER DISTRIBUTES A BUFFERED VERSION TO THE PXB AND BACK INTO THE CONTROLLER TO THE PLL FEEDBACK. THERE IS ONE OF THESE FOR EACH BUS.
XCLK	EXT. CLK DRIVER	PXB	THE BUFFERED VERSION OF THE XCLKOUT FROM THE EXTERNAL LOW SKEW CLOCK DRIVER SENT TO THE PXB. THERE IS ONE OF THESE FOR EACH BUS.
XCLK_IN	EXT. CLK DRIVER	CONTROLLER	THE BUFFERED VERSION OF THE XCLKOUT FROM THE EXTERNAL LOW SKEW CLOCK DRIVER SENT TO THE CONTROLLER FOR THE XCLK PLL FEEDBACK. THERE IS ONE OF THESE FOR EACH BUS.
RESET	CONTROLLER	PXB	SYSTEM RESET GENERATED BY THE CONTROLLER.
XRST #	CONTROLLER	PXB	PXB RESET GENERATED BY THE CONTROLLER. THERE IS A SEPARATE ONE FOR EACH PXB. THE CONTROLLER AND PXB SOURCE SYNCHRONOUS DATA STROBES START RUNNING ON THE FIRST CLOCK AFTER OBSERVING XRST # DEASSERT.
XRSTIN #	CONTROLLER	CONTROLLER	A COPY OF XRST # FED BACK INTO THE CONTROLLER SO THAT THE PXB AND CONTROLLER SEE XRST # DEASSERT ON THE SAME CLOCK.

FIG. 2B

SIGNAL	SRC	DEST.	DEFINITION
X_P_STB#	PXB	CONTROLLER	PCI EXPANDER POSITIVE DATA STROBE SOURCES SYNCHRONOUS CLOCK STROBE SENT FROM PXB TO CONTROLLER WITH POSITIVE PHASE DATA. THIS STROBE CLOCK STARTS RUNNING ON THE FIRST CLOCK AFTER OBSERVING XRST# DEASSERT. THE STROBE IS FREE-RUNNING UNTIL ANOTHER ASSERTION OF XRST#.
X_N_STB#	PXB	CONTROLLER	PCI EXPANDER NEGATIVE DATA STROBE SOURCE SYNCHRONOUS CLOCK STROBE SENT FROM PXB TO CONTROLLER WITH NEGATIVE PHASE DATA. THIS STROBE CLOCK STARTS RUNNING ON THE FIRST CLOCK AFTER OBSERVING XRST# DEASSERT. THE STROBE IS FREE-RUNNING UNTIL ANOTHER ASSERTATION OF XRST#.
B_P_STB#	CONTROLLER	PXB	CONTROLLER POSITIVE DATA STROBE SOURCE SYNCHRONOUS CLOCK STROBE SENT FROM CONTROLLER TO PXB WITH POSITIVE PHASE DATA. THIS STROBE CLOCK STARTS RUNNING ON THE FIRST CLOCK AFTER OBSERVING XRST# DEASSERT. THE STROBE IS FREE - RUNNING UNTIL ANOTHER ASSERTION OF XRST#.
B_N_STB#	CONTROLLER	PXB	CONTROLLER NEGATIVE DATA STROBE SOURCE SYNCHRONOUS CLOCK STROBE SENT FROM CONTROLLER TO PXB WITH NEGATIVE PHASE DATA. THIS STROBE CLOCK STARTS RUNNING ON THE FIRST CLOCK AFTER OBSERVING XRSTIN# DEASSERT. THE STROBE IS FREE - RUNNING UNTIL ANOTHER ASSERTATION OF XRST#.
XRST #	PXB	CONTROLLER	PCI EXPANDER REQUEST TO SEND REQUEST TO USE THE BI - DIRECTIONAL BUS SENT FROM PXB TO CONTROLLER SYNCHRONOUS TO XCLK#.

FIG. 2C

SIGNAL	SRC	DEST.	DEFINITION
HRST#	CONTROLLER	PXB	CONTROLLER REQUEST TO SEND REQUEST TO USE THE BI - DIRECTIONAL BUS SENT FROM CONTROLLER TO PXB SYNCHRONOUS TO XCLK..
XADS#	CONTROLLER / PXB	CONTROLLER / PXB	PCI EXPANDER ADDRESS/DATA STROBE BI - DIRECTIONAL SIGNAL ASSERTED BY THE SENDING AGENT DURING EVERY CLOCK OF A PACKET TRANSMISSION EXCEPT THE LAST CLOCK. IN A SINGLE CLOCK TRANSMISSION SUCH AS A COMPLETION PACKET WITH NO DATA, X_ADS# IS ASSERTED FOR ONE CLOCK.
XBE[1:0]#	CONTROLLER / PXB	CONTROLLER / PXB	PCI EXPANDER BYTE ENABLES BI - DIRECTIONAL SIGNALS ASSERTED IN PHASE WITH DATA ON THE BUS TO INDICATE VALID BYTES DURING THE DATA PHASES OF A PACKET TRANSMISSION. RESERVED FUNCTION DURING HEADER PHASES.

FIG. 2D

SIGNAL	SRC	DEST.	DEFINITION
XD[15:0]#	CONTROLLER / PXB	CONTROLLER / PXB	PCI EXPANDER DATA BUS BI - DIRECTIONAL BUS TRANSFERS PACKETS BETWEEN THE CONTROLLER AND PXB.
XADS#	CONTROLLER / PXB	CONTROLLER / PXB	PCI EXPANDER BUS PARITY BI - DIRECTIONAL SIGNAL INDICATES EVEN PARITY ACROSS XD[15:0] AND XBE[1:0]. MUST BE VALID FOR EVERY HALF CLOCK PHASE OF A PACKET TRANSMISSION. INVALID PARITY WILL BE GENERATED FOR DATA THAT HAD UNCORRECTABLE MEMORY OR BUS ERRORS. A PARITY SIGNAL IS CORRECT IF THERE ARE AN EVEN NUMBER OF ELECTRICALLY LOW SIGNALS IN THE SET INCLUDING THE COVERED SIGNALS PLUS THE PARITY SIGNALS.

FIG. 2E

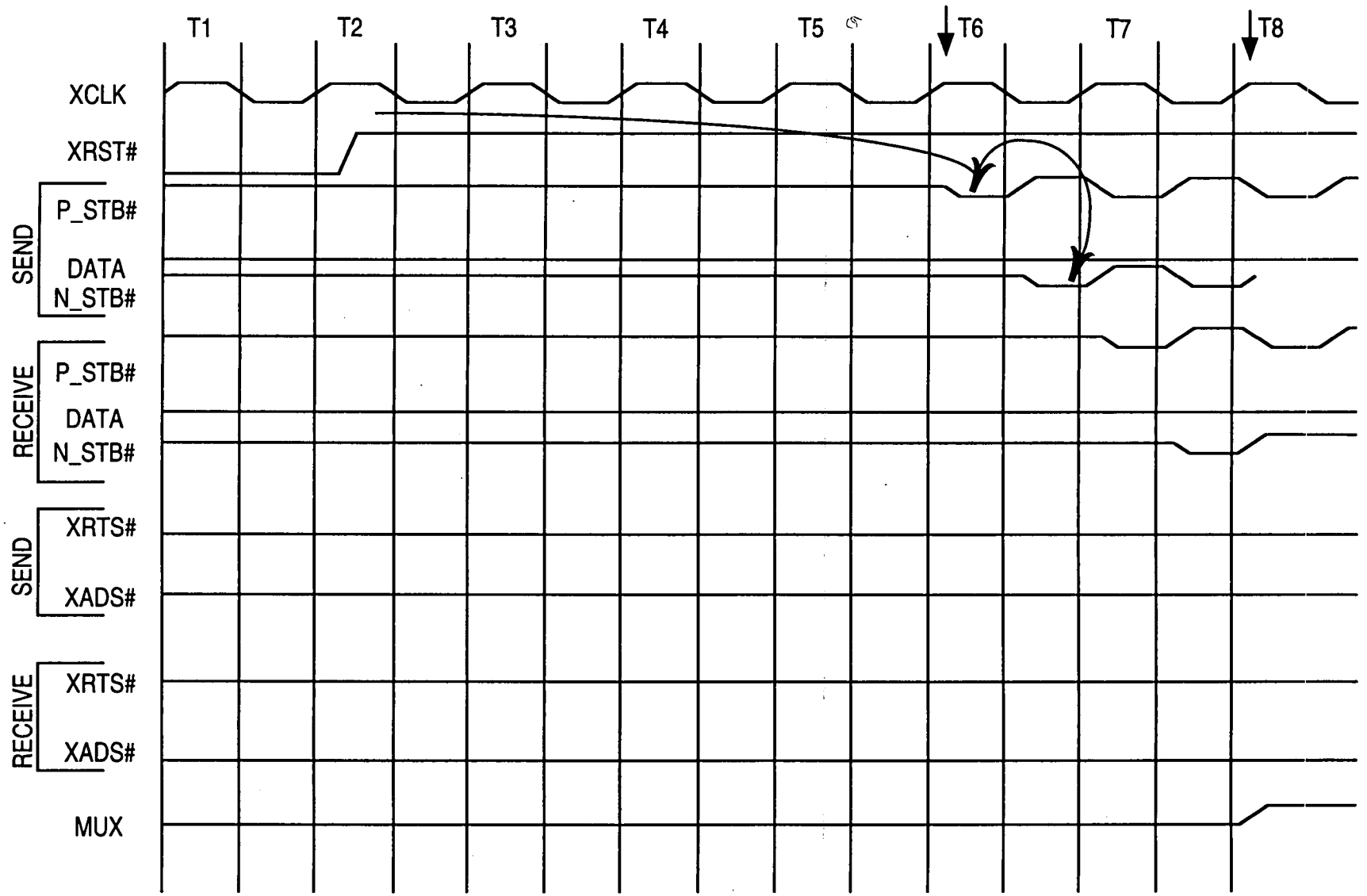

FIG 3: F16 STROBE STARTUP TIMING DETAIL

FIG. 3

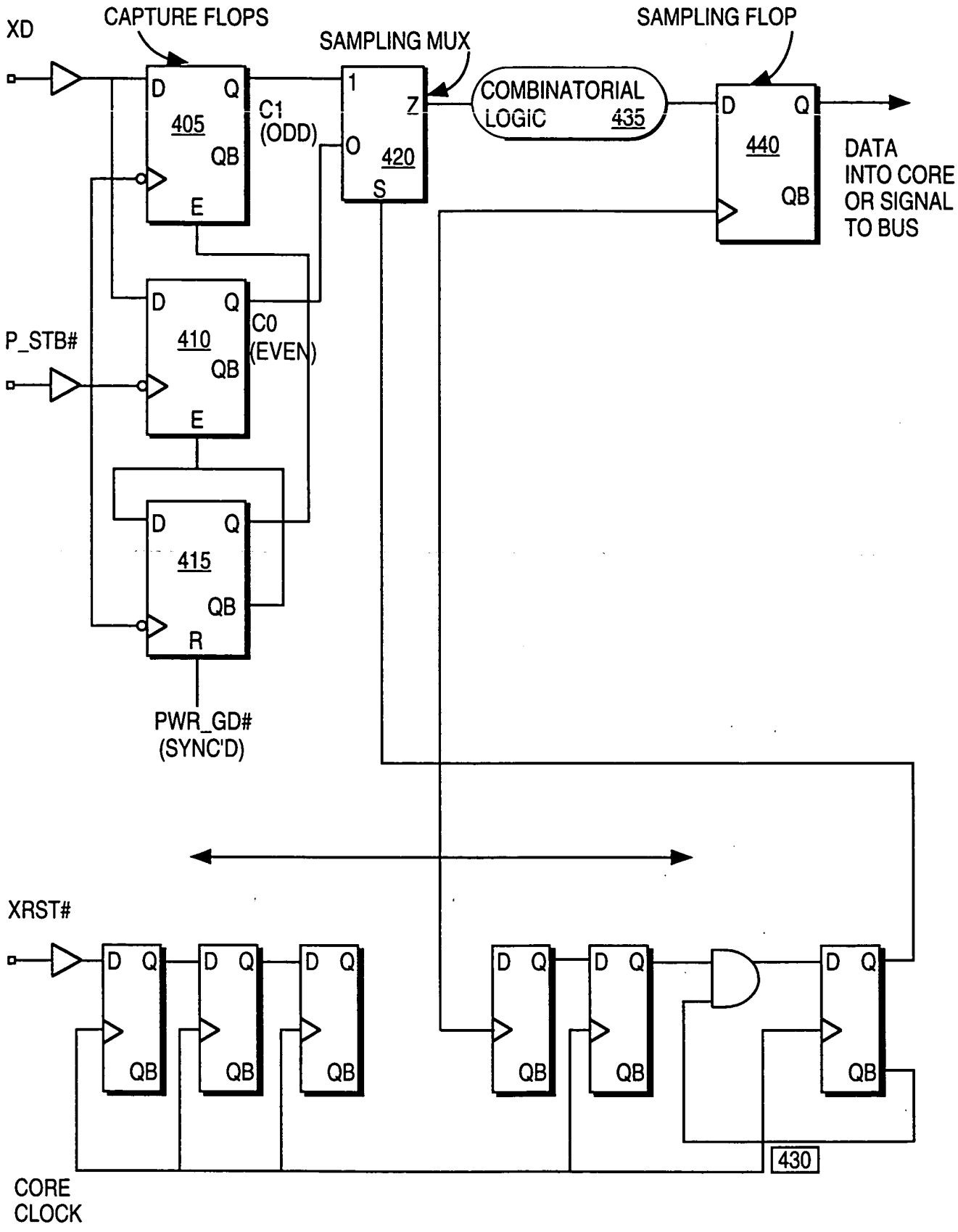
XD
CAPTURE FLOPS
SAMPLING MUX
SAMPLING FLOP
D
Q
405
QB
E
C1
(ODD)
1
Z
O
420
S
COMBINATORIAL
LOGIC
435
440
DATA
INTO CORE
OR SIGNAL
TO BUS
P_STB#
410
C0
(EVEN)
415
R
PWR_GD#
(SYNC'D)
XRST#
430
CORE
CLOCK

FIG. 4

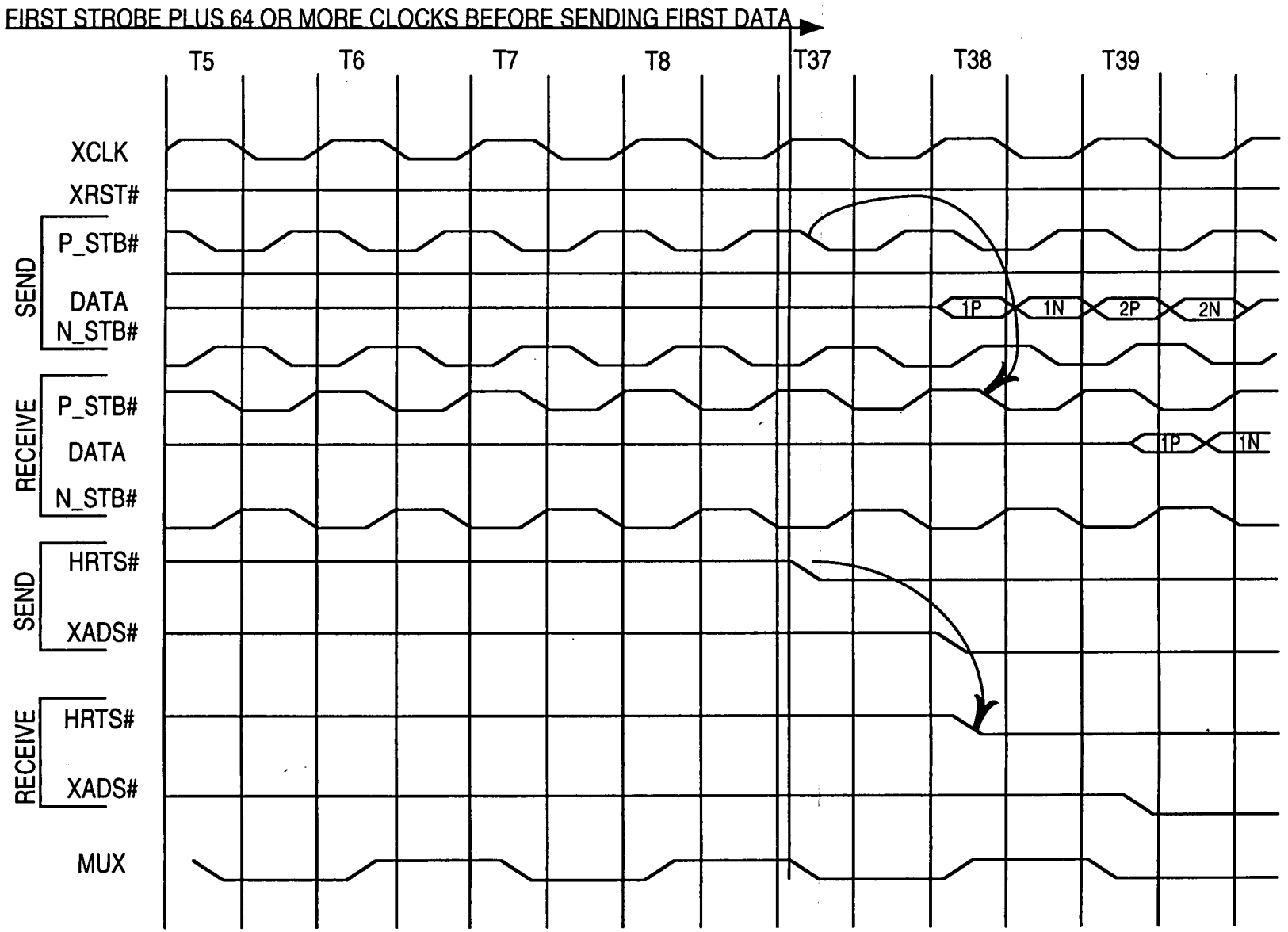

FIGURE 5: F16 DATA TRANSMISSION DETAIL

FIG. 5

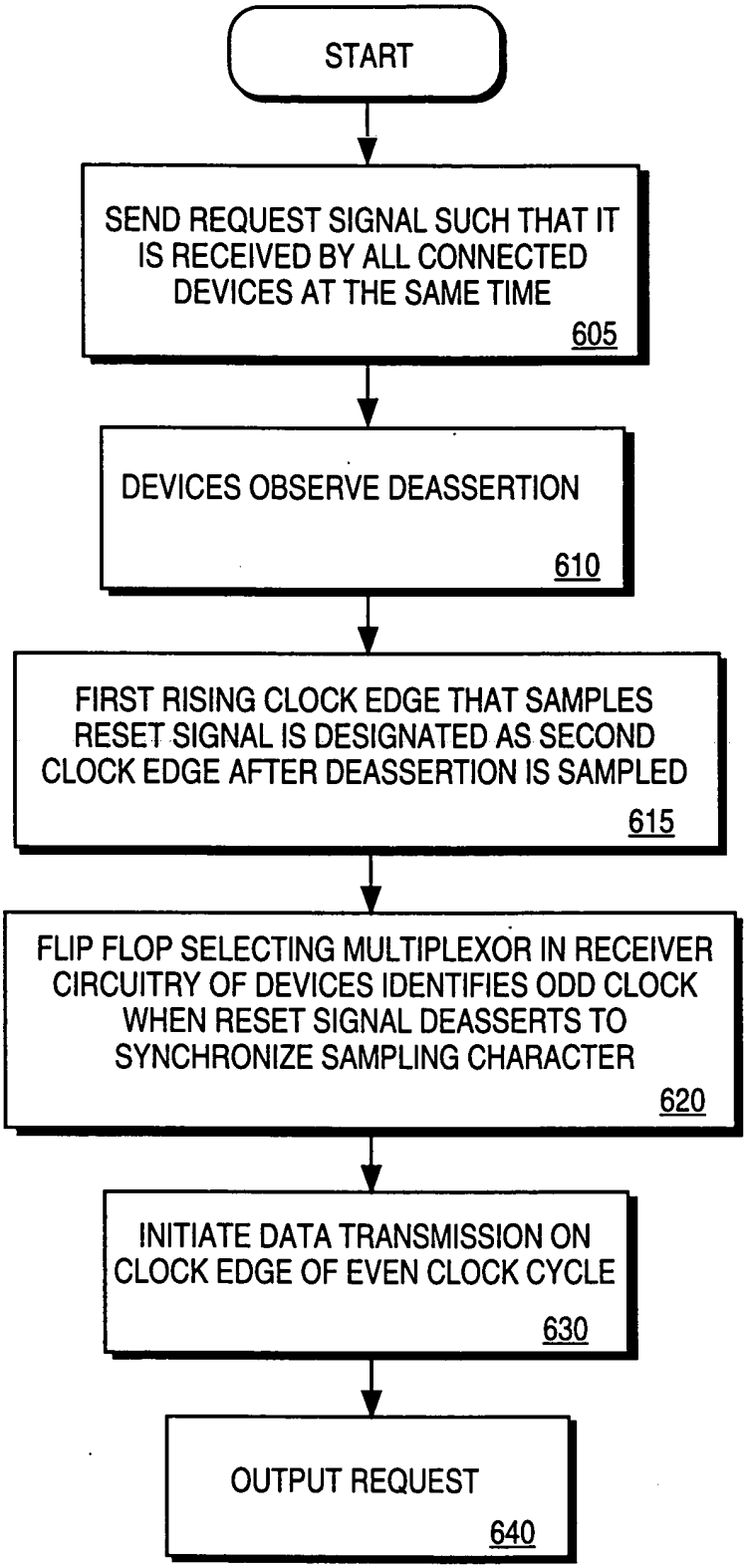
START
SEND REQUEST SIGNAL SUCH THAT IT IS RECEIVED BY ALL CONNECTED DEVICES AT THE SAME TIME
605
DEVICES OBSERVE DEASSERTION
610
FIRST RISING CLOCK EDGE THAT SAMPLES RESET SIGNAL IS DESIGNATED AS SECOND CLOCK EDGE AFTER DEASSERTION IS SAMPLED
615
FLIP FLOP SELECTING MULTIPLEXOR IN RECEIVER CIRCUITRY OF DEVICES IDENTIFIES ODD CLOCK WHEN RESET SIGNAL DEASSERTS TO SYNCHRONIZE SAMPLING CHARACTER
620
INITIATE DATA TRANSMISSION ON CLOCK EDGE OF EVEN CLOCK CYCLE
630
OUTPUT REQUEST
640

FIG. 6

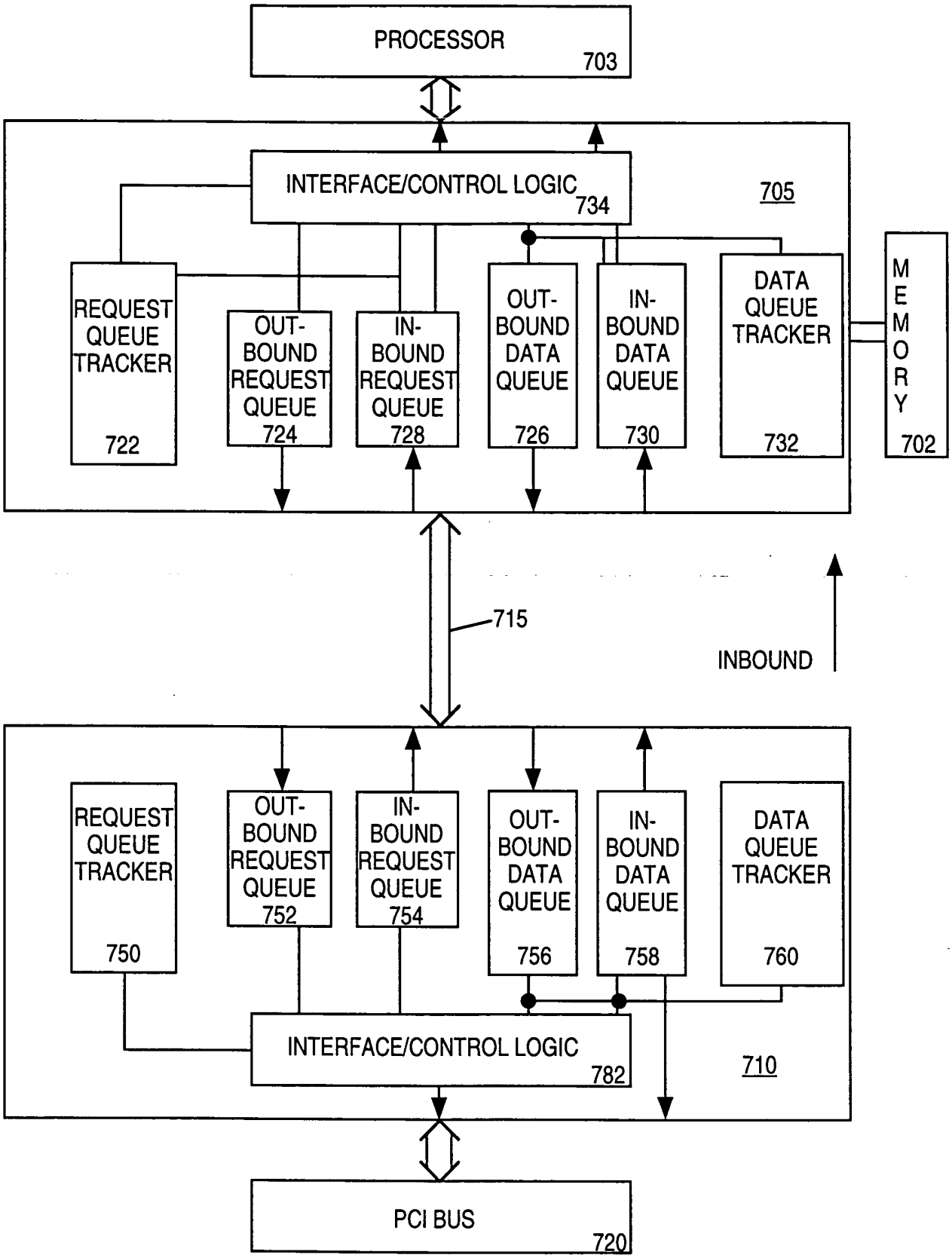
PROCESSOR
703
INTERFACE/CONTROL LOGIC
734
705
REQUEST QUEUE TRACKER
722
OUT-BOUND REQUEST QUEUE
724
IN-BOUND REQUEST QUEUE
728
OUT-BOUND DATA QUEUE
726
IN-BOUND DATA QUEUE
730
DATA QUEUE TRACKER
732
MEMORY
702
715
INBOUND
REQUEST QUEUE TRACKER
750
OUT-BOUND REQUEST QUEUE
752
IN-BOUND REQUEST QUEUE
754
OUT-BOUND DATA QUEUE
756
IN-BOUND DATA QUEUE
758
DATA QUEUE TRACKER
760
INTERFACE/CONTROL LOGIC
782
710
PCI BUS
720

FIG. 8

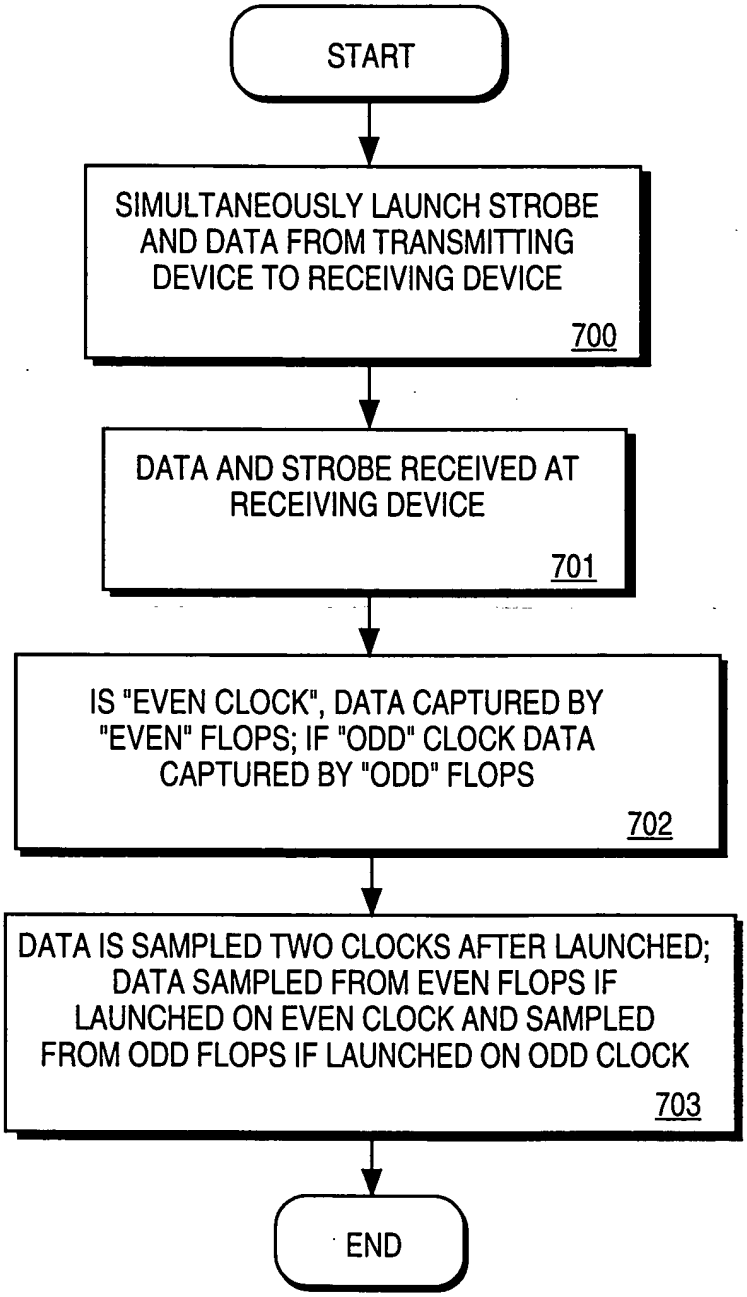
START
SIMULTANEOUSLY LAUNCH STROBE AND DATA FROM TRANSMITTING DEVICE TO RECEIVING DEVICE
700
DATA AND STROBE RECEIVED AT RECEIVING DEVICE
701
IS "EVEN CLOCK", DATA CAPTURED BY "EVEN" FLOPS; IF "ODD" CLOCK DATA CAPTURED BY "ODD" FLOPS
702
DATA IS SAMPLED TWO CLOCKS AFTER LAUNCHED; DATA SAMPLED FROM EVEN FLOPS IF LAUNCHED ON EVEN CLOCK AND SAMPLED FROM ODD FLOPS IF LAUNCHED ON ODD CLOCK
703
END

FIG. 7

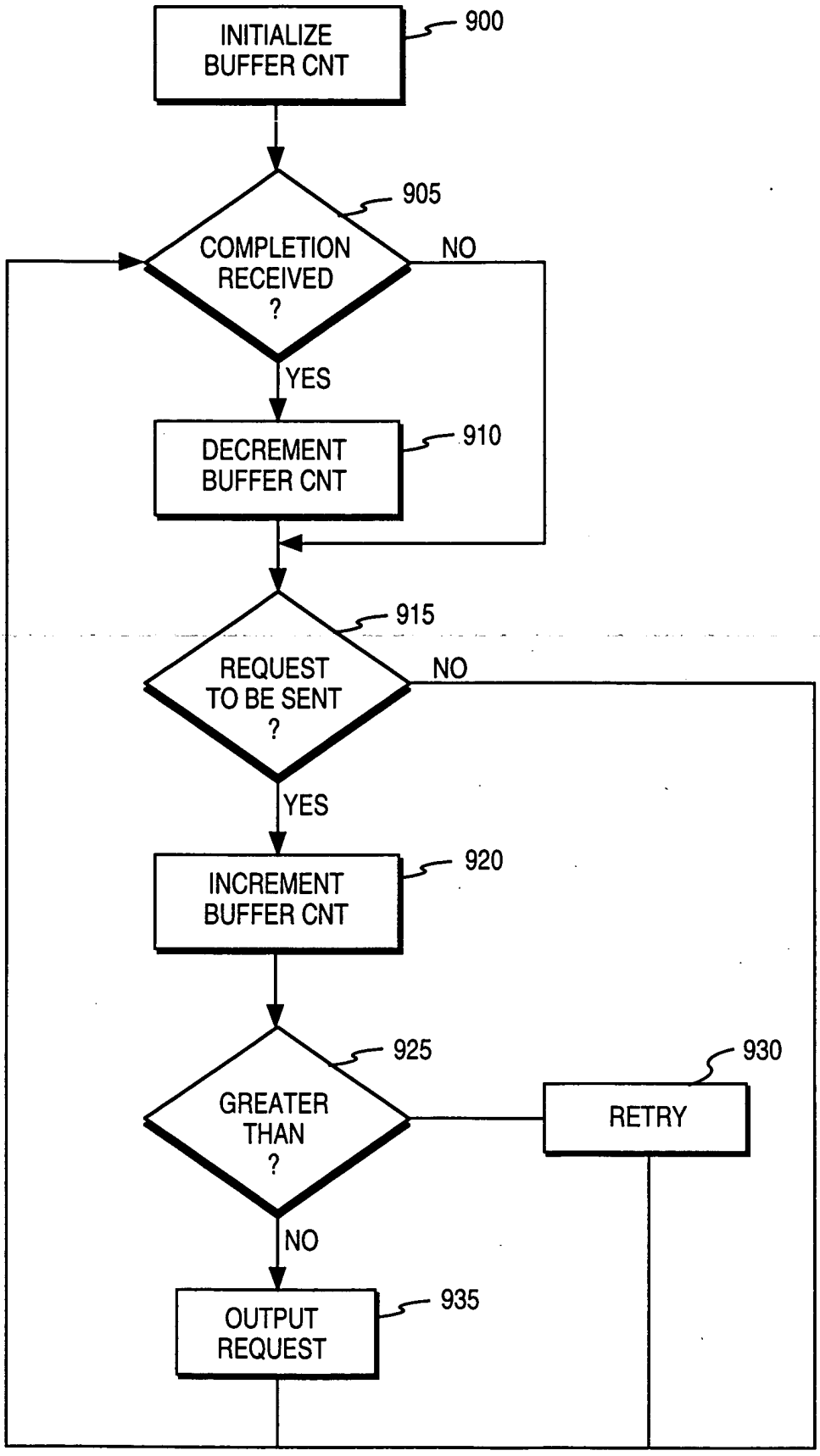
INITIALIZE BUFFER CNT
900
COMPLETION RECEIVED ?
905
NO
YES
DECREMENT BUFFER CNT
910
REQUEST TO BE SENT ?
915
NO
YES
INCREMENT BUFFER CNT
920
GREATER THAN ?
925
RETRY
930
NO
OUTPUT REQUEST
935

FIG. 9A

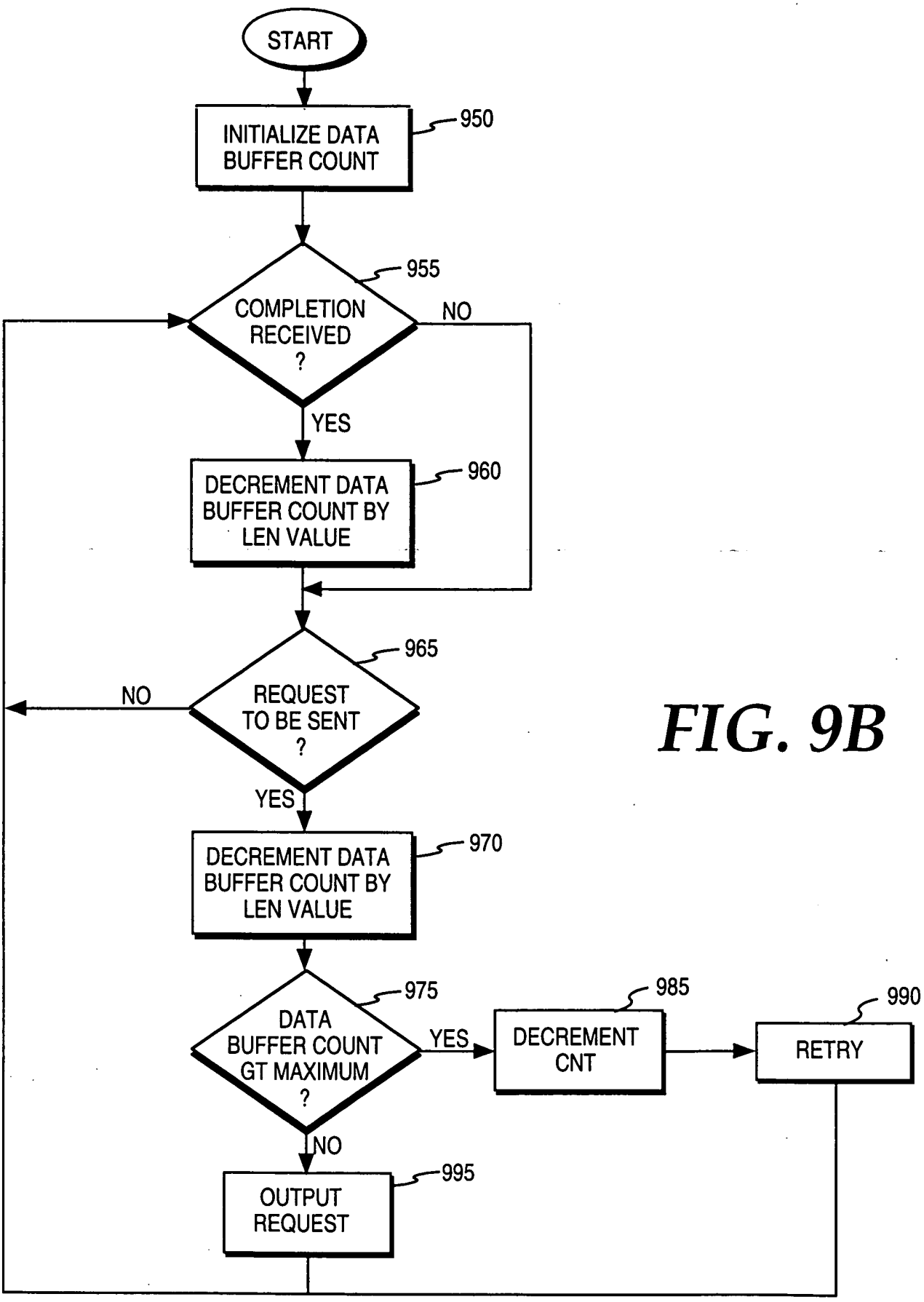
START
INITIALIZE DATA BUFFER COUNT
950
COMPLETION RECEIVED ?
955
NO
YES
DECREMENT DATA BUFFER COUNT BY LEN VALUE
960
REQUEST TO BE SENT ?
965
NO
YES
DECREMENT DATA BUFFER COUNT BY LEN VALUE
970
DATA BUFFER COUNT GT MAXIMUM ?
975
YES
DECREMENT CNT
985
RETRY
990
NO
OUTPUT REQUEST
995

FIG. 9B

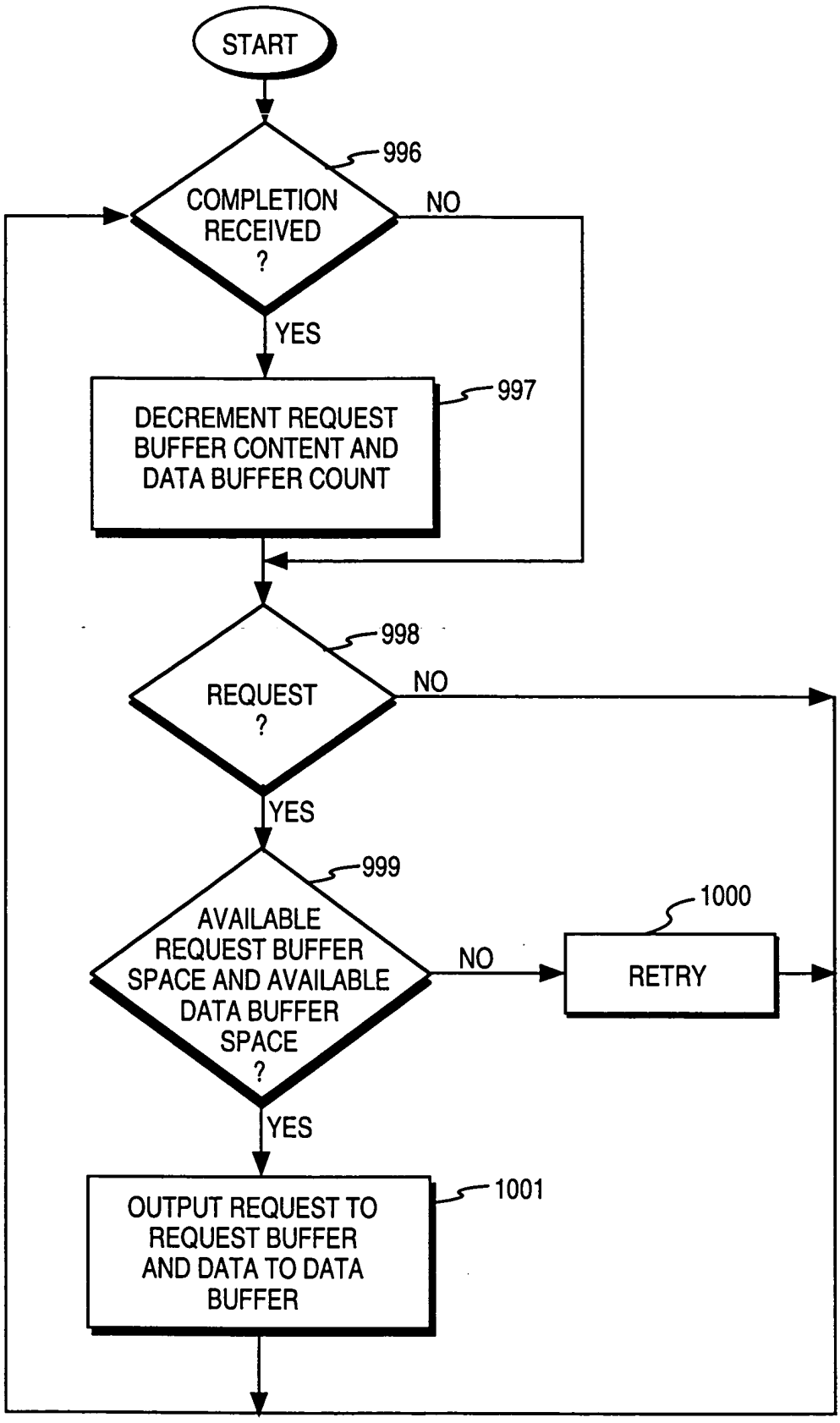
START
996
COMPLETION RECEIVED ?
NO
YES
997
DECREMENT REQUEST BUFFER CONTENT AND DATA BUFFER COUNT
998
REQUEST ?
NO
YES
999
AVAILABLE REQUEST BUFFER SPACE AND AVAILABLE DATA BUFFER SPACE ?
NO
1000
RETRY
YES
1001
OUTPUT REQUEST TO REQUEST BUFFER AND DATA TO DATA BUFFER

FIG. 9C

REQUEST PACKET FORMAT										
CLK CYCLE	**BE[1:0] TP[1:0]**	**XD[15]**	**XD[14:12]**	**XD[11]**	**XD[10]**	**XD[9:8]**	**XD[7]**	**XD[6:4]**	**XD[3:2]**	**XD[1:0]**
1P	0,X	LH=1	ROUTE[3:0]		CA[5:2]			RCOM[4:0]		TID[9:8]
1N	TID[7:0]					LEN[7:0]				RBE[7:6]
2P	RBE[5:0]			A[35:24]						
2N	A[23:6]									
3P	RESERVED						A[43:36]			
3N	RESERVED						RESERVED			
4P	BE[3:2]	DATA[31:24]					DATA[23:16]			
4N	BE[1:0]	DATA[15:8]					DATA[7:0]			

FIG. 10A

COMPLETION PACKET FORMAT									
F16 CLK CYCLE	**BE[1:0] TP[1:0]**	**XD[15]**	**XD[14:12]**	**XD[11]**	**XD[10]**	**XD[9:7]**	**XD[6:4]**	**XD[3:2]**	**XD[1:0]**
1P	1,X	NA	ROUTE[3:0]		CA[5:2]		CCOM[4:0]		TID[9:8]
1N	TID[7:0]					LEN[7:0]			RBE[7:6]
2P	BE[3:2]	DATA[31:24]					DATA[23:16]		
2N	BE[1:0]	DATA[15:8]					DATA[7:0]		

FIG. 10B

PACKET FIELDS	
COMMAND	**DESCRIPTION**
RCOM (COMMANDS)	INTERRUPT ACKNOWLEDGE; SPECIAL TRANSACTION; I/O READ; MEMORY READ; MEMORY WRITE; CHECK CONNECTION; UNLOCK; CONFIGURATION READ; CONFIGURATION WRITE; SSBR; RSBR; LOCKED READ; MEMORY WRITE LINE
CCOM[4:0] (COMPLETION COMMANDS)	RETURNS STATUS OF ASSOCIATED REQUEST: NORMAL READ; NORMAL WRITE; RETRY READ; RETRY WRITE (MAP TO HARD FAIL); HARD FAIL READ; HARD FAIL WRITE; SHORT PACKET READ; SHORT PACKET WRITE (MAP TO HARD FAIL)
TID[9:0] (TRANSACTION ID)	"SCRATCH PAD" CREATED BY REQUESTOR - UNMODIFIED BY TARGET. USED BY REQUESTOR TO TRACK COMPLETIONS AS THEY RETURN.
LEN[7:0] (DATA LENGTH IN DWORDS)	INDICATES NUMBER OF DWORD REQUESTED AND NUMBER OF DWORDS RETURNED OF TO DE-ALLOCATE; DATA BUFFER TRACKER ADDS THIS NUMBER TO THE WRITE BUFFER POOL WHEN A WRITE RETURNS
RBE[7:0] (REQUEST BYTE ENABLES)	ACTIVE BYTES FOR REQUESTS OF 8 BYTES OR LESS; RBE[7:4] IS ALWAYS THE ODD DWORD BE'S; RBE[3:0] IS ALWAYS THE EVEN DWORD BE'S
A[43:36} (REQUEST ADDRESS)	A[43:36] IN LONG HEADER FORMAT ONLY; SHORT HEADER ONLY BEING VALIDATED ON 450 NX. 460 GX POR DOES NOT USE LONG HEADER
SSBR (SECURE SIDEBAND REQUEST)	ISSUED BY PXB TO MIOC WHEN PHOLD ASSERTS; MIOC BLOCKS FURTHER OUTBOUND REQUESTS TO COMPATIBILITY PCI BUS
RSBR (REMOVE SIDEBAND REQUEST)	ISSUED BY PBX TO MIOC WHEN PHOLD DEASSERTS - NOT PASSIVE RELEASE

FIG. 10C

PACKET FIELDS	
COMMAND	**DESCRIPTION**
LOCKED READ	ISSUED BY MIOC TO PBX FOR THE FIRST READ OF A LOCKED SEQUENCE; MIOC BLOCKS FURTHER REQUESTS TO THE TARGETED PCI BUS EXCEPT BY THE AGENT INITIATING THE LOCK
UNLOCK	ISSUED BY MIOC TO PBX WHEN FSB LOCK NUMBER SIGNAL DEASSERTS

FIG. 10D

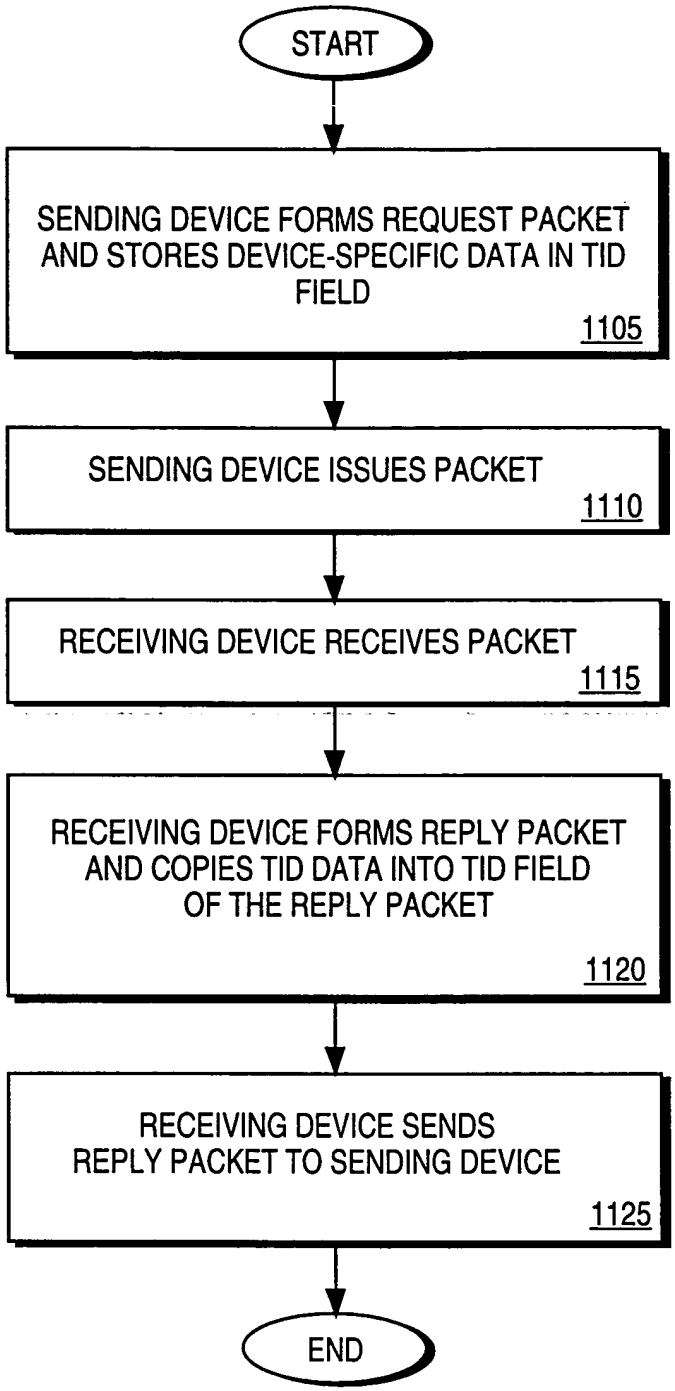
START
SENDING DEVICE FORMS REQUEST PACKET AND STORES DEVICE-SPECIFIC DATA IN TID FIELD
1105
SENDING DEVICE ISSUES PACKET
1110
RECEIVING DEVICE RECEIVES PACKET
1115
RECEIVING DEVICE FORMS REPLY PACKET AND COPIES TID DATA INTO TID FIELD OF THE REPLY PACKET
1120
RECEIVING DEVICE SENDS REPLY PACKET TO SENDING DEVICE
1125
END

FIG. 11